基于SDK ADC-DMA多通道采样-KE15

基于SDK ADC-DMA多通道采样-KE15

在使用Kinetis系列的时候，经常有客户问，如何采样多通道ADC值，而无需频繁的产生中断。
主要是Kinetis系列ADC与LPC/DSC及某些其他MCU相比，缺少了ADC通道序列采样的功能。即只在初始化时指定需要转化的多个ADC通道号，使能一次转化即可在完成所有ADC通道采样后产生中断，并在中断中读取所有采样数据。

虽然Kinetis ADC没有这个功能，但是借助强大的DMA，还是能够实现这个功能，并且能够提供更大的灵活性，对于需要更灵活的使用ADC采样值的应用，更有意义。

下面介绍多种ADC+DMA配合的例子，按需获取即可。

基于SDK ADC-DMA多通道采样-KE15

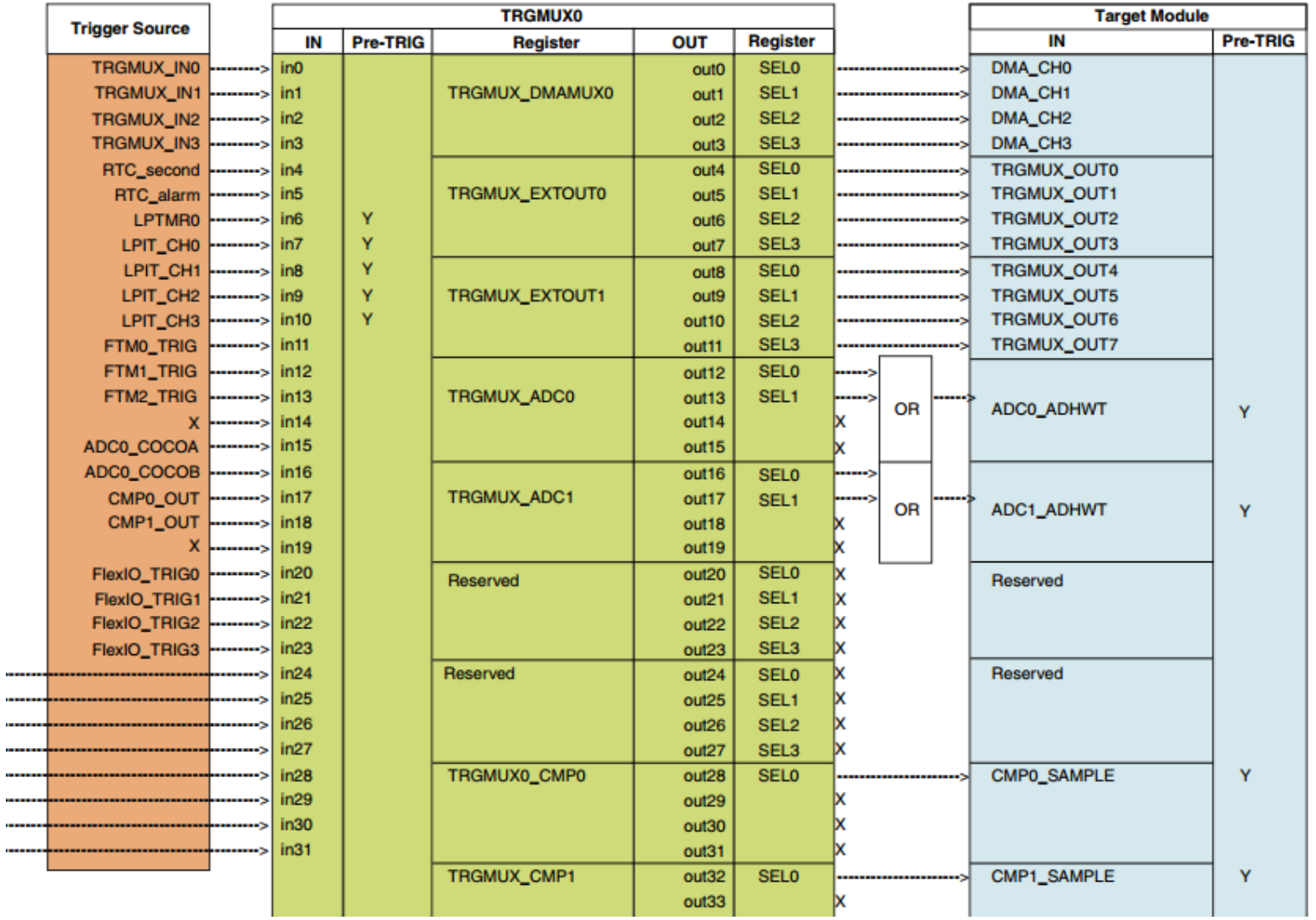
基本概念

基于SDK ADC-DMA多通道采样-KE15-ADC

- KE15拥有两个ADC模块，且每个ADC模块有两个结果寄存器，如果合理的使用Pre-trigger/Trigger就可以在一个ADC模块上连续产生两个通道的ADC采样值。使用TRGMUX可以为ADCx生成Pre-trigger/Trigger信号。比如使用LPIT和FTM定时器可以在特定的时间内产生ADC触发信号。

- 对于多通道的ADC转换，以上两个结果寄存器显然还不够，需要使用软件触发的方式。每一次ADCx_SC1A寄存器的ADCH被写一次就触发一次软件触发转换。转换完成后既可以生成中断，也可以作为DMA的触发源，触发一次DMA转换。.

- 第二个方法是我们准备使用的模式。



基于SDK ADC-DMA多通道采样-KE15-DMAMUX

- Kinetis的DMA功能非常强大，能够通过软件或者硬件外设触发。管理触发源的外设称之为DMA MUX。触发源包括了通信外设（串口/SPI/I2C等），GPIO变化，定时器超时事件，软件触发等等。

- 注意：KE15拥有TRGMUX，可以通过TRGMUX的TRGMUX_DMAMUX out0-3直接触发DMA 0-3通道，这是比较特殊的一点。比如可以使用LPIT超时信号route到TRGMUX_DMAMUX out0直接触发DMA_CH0.

- 如果在早期没有TRGMUX的芯片上（K6X等），也有类似使用PIT“周期性触发”DMA的功能。

- 即使不使用LPIT触发，DAMMUX中也有LPTMR和FTM等定时器资源可以用于周期性触发。

Table 12-1. DMA request sources - MUX 0 (continued)

Source number	Source module	Source description	Async DMA capable
50	Port control module	Port B	Yes
51	Port control module	Port C	Yes
52	Port control module	Port D	Yes
53	Port control module	Port E	Yes
54	Reserved	—	
55	Reserved	—	
56	Reserved		
57	FTM1	OR of ch2-ch3	
58	FTM2	OR of ch2-ch3	
59	LPTMR0	—	Yes
60	DMAMUX	Always enabled	
61	DMAMUX	Always enabled	
62	DMAMUX	Always enabled	
63	DMAMUX	Always enabled	

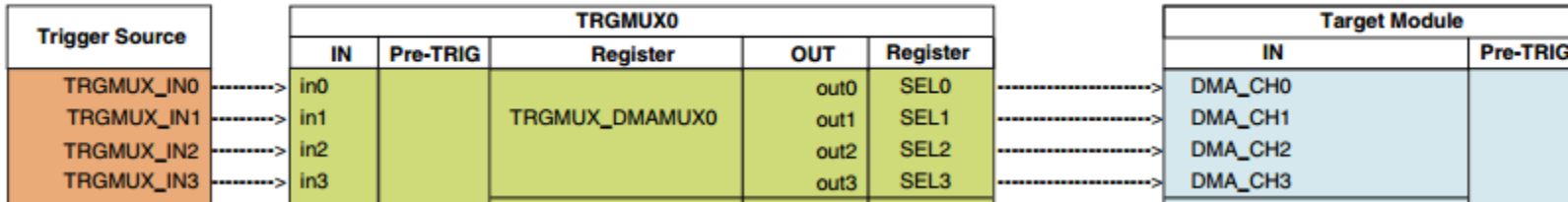
1. Configuring a DMA channel to select source 0 or any of the reserved sources disables that DMA channel.

12.1.2 DMA trigger sources

The DMAMUX on this device also supports a periodic trigger mode. The trigger sources are from TRGMUX output showed in following table. The triggers from TRGMUX module can trigger a DMA transfer on the first four DMA channels (channel 0 -3), for example, the LPIT can trigger DMA via TRGMUX.

Table 12-2. DMAMUX trigger sources

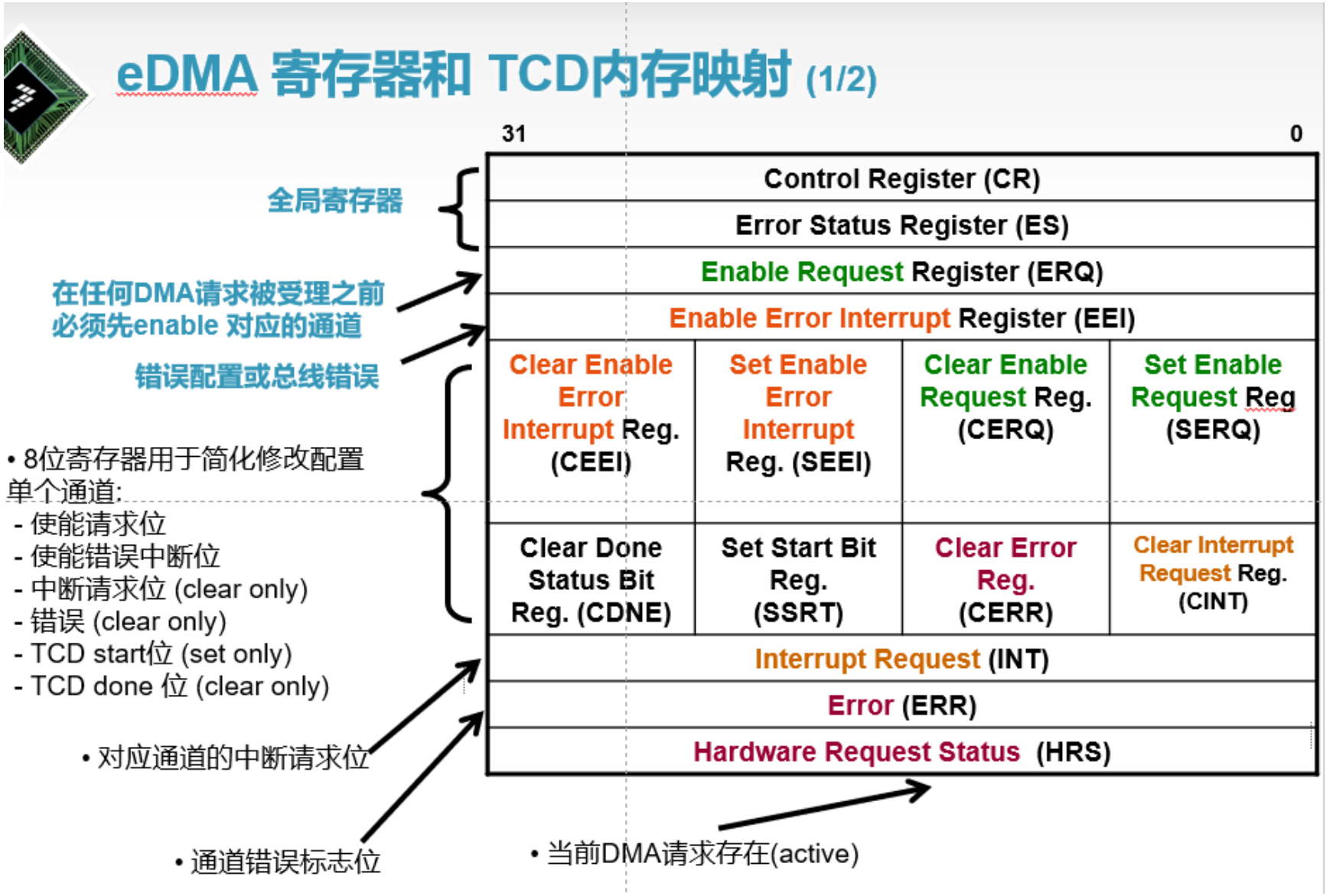
Trigger number	Trigger module	Trigger description
0	TRGMUX	TRGMUX trigger out0
1	TRGMUX	TRGMUX trigger out1
2	TRGMUX	TRGMUX trigger out2
3	TRGMUX	TRGMUX trigger out3



基于SDK ADC-DMA多通道采样-KE15-DMA

DMA寄存器中比较重要的寄存器包括：

1. CR寄存器有关于对DMA总括性的设置；
2. 当DMA传输出现错误，ES寄存器可以观察到错误原因；
3. ERQ/SERQ/CERQ寄存器用于设置当前通道触发信号是否允许触发该通道DMA；
4. EEI/SEEI/CEEI用于设置某通道DMA错误是否可以产生中断；
5. ERR寄存器可以看到哪个通道产生了错误。
6. CDNE/CERR/CINT都是用于清除某单一DMA通道的相应标志位；
7. HRS硬件触发标志



基于SDK ADC-DMA多通道采样-KE15-DMA TCD

TCD：可以使用一块32Bytes的memory来初始化对应的DMA通道设置寄存器。包括源地址，目的地址，每次传输的size，总长度，偏移地址。传输完成后跳转等一系列设置。第一次使用TCD memory设置对应寄存器，并在DLAST_SGA处设置下一个TCD的指针地址，这样在第一次DMA传输完成后，会自动根据下一次的TCD配置DMA寄存器。无需在中断中手动配置。

- 在DMA请求被服务之前， 对应通道的TCD必须被正确配置.
- TCD (Transfer Control Descriptors) 是位于本地memory 的数据结构，用来描述如何来处理DMA请求
- 每个DMA通道对应一个长32bytes TCD

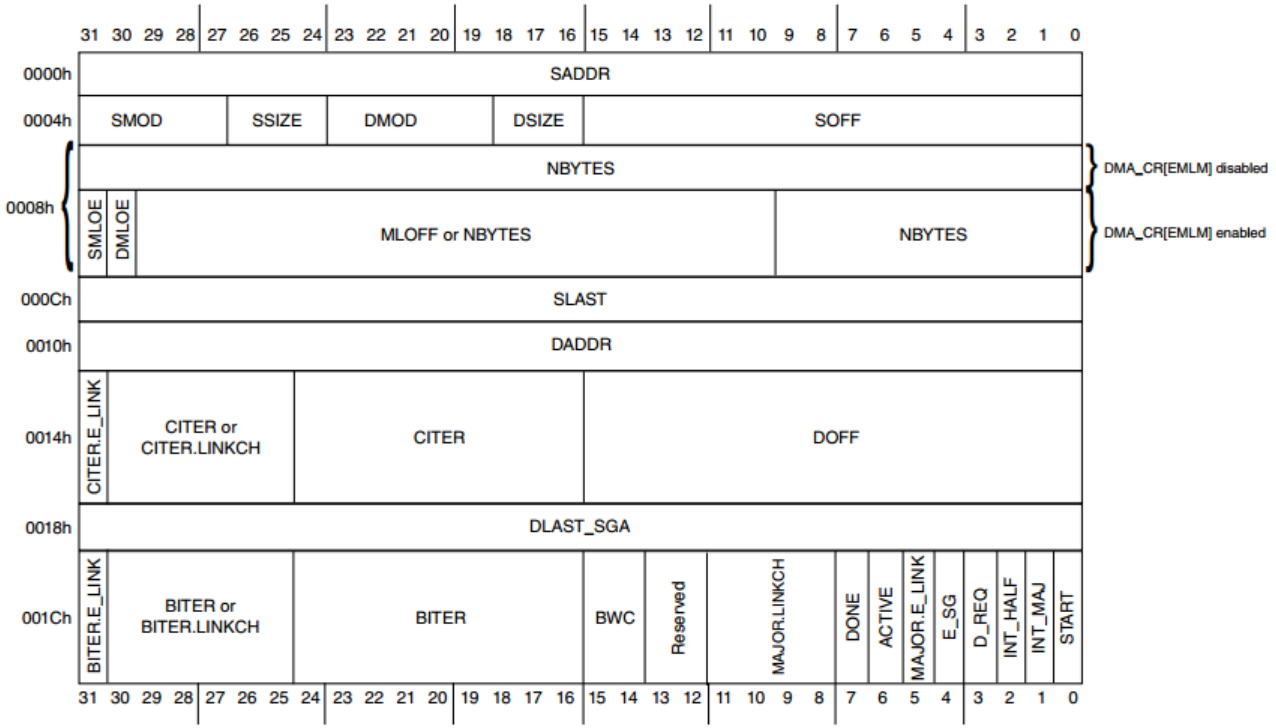
TCD Memory Map

eDMA Offset	TCDn Register Name	Abbreviation	Width (bits)
0x00	源地址 （Source Address）	TCDn_SADDR	32
0x04	源地址偏移，有符号数（Signed Source Address Offset）	TCDn_SOFF	16
0x06	传送属性（Transfer Attributes）	TCDn_ATTR	16
0x08	内循环字节 （Minor Byte Count）	TCDn_NBYTES	32
0x0C	源地址后调整（Last Source Address Adjustment）	TCDn_SLAST	32
0x10	目标地址 （Destination Address）	TCDn_DADDR	32
0x14	目标地址偏移，有符号（Signed Destination Address Offset）	TCDn_DOFF	16
0x16	当前内循环链接，主循环计数（Current Minor Loop Link, Major Loop Count）	TCDn_CITER	16
0x18	目的地址后调整（Last Destination Address Adjustment/**Scatter Gather** Address）	TDDn_DLAST_SGA	32
0x1C	控制和状态（Control and Status）	TCDn_CSR	16
0x1E	Beginning Minor Loop Link, Major Loop Count	TCDn_BITER	16

13.3.1 TCD memory

Each channel requires a 32-byte transfer control descriptor for defining the desired data movement operation. The channel descriptors are stored in the local memory in sequential order: channel 0, channel 1, ... channel 7. Each TCD*n* definition is presented as 11 registers of 16 or 32 bits.

13.3.3 TCD structure



基于SDK ADC-DMA多通道采样-KE15-DMA&DMAMUX

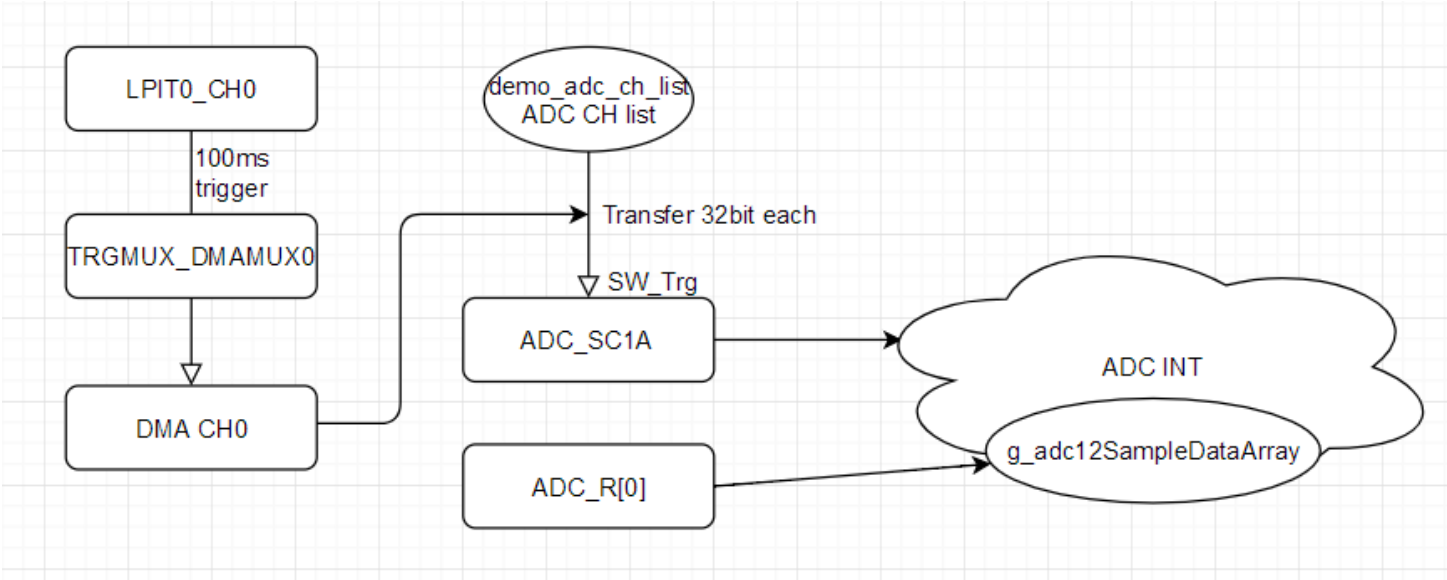
关于DMA/TCD
的详细介绍见
eDMA模块介绍，
请看这个PPT

MULTI_CHANNELS_EDMA_ADC_INTERRUPT

multi_channels_edma_ADC_interrupt

此项目为一系列ADC-DMA工程的第一步，完成的功能包括：

1. 建立一个多通道的ADC channel数组demo_adc_ch_list。分别为ADC0_ch0(PTA0), temp sensor, bandgap, vrefh, vrefl, ADC0_ch0(PTA0)。
2. ADC_USING_DMA_INT设置为0，ADC软件触发产生中断，在中断中取得该通道转换结果。
3. 使用DMA ch0将此ADC通道号依次传送到ADC0_SC1A，软件触发ADC转换。为使ADC在每次转换完成后产生中断，以上通道号数组需要置上AIEN bit；
4. DMA ch0由LPIT ch0通过TRGMUX来周期触发DMA ch0（详见手册periodic triggering及trgmux的章节。LPIT的ch0-3可以通过TRGMUX_DMAMUX0触发DMAMUX_ch0-3）
5. DMA ch0设置为单次传输4字节，总共分6次传输。完成6个通道的转换。每次LPIT超时触发一次。所以DMA有minor loop和major loop，每次minor loop就一次4字节，总共6次major loop
6. 每次ADC转换完成都会进入中断，并保存ADC转换结果
7. DMA转换major loop也设置中断，完成后打印所有ADC转换结果。



这样完成了一个使用LPIT定时触发多个通道的示例，多个通道采样的间隔时间即为LPIT的超时时间。

MULTI_CHANNELS_EDMA_ADC_DMA_LPIT_HW_TRIG

multi_channels_edma_ADC_DMA_LPIT_HW_TRIG

有些应用不希望像以上示例一样，ADC的多通道是指定好的间隔采样。而是希望每间隔一段时间，连续采样多通道，接着下一次循环这个过程。

此项目在上面项目的基础上，新增：

1.ADC的转换结果不会产生中断，而是触发DMA ch1去搬运到g_adc12SampleDataArray数组中；

2.DMA ch1的minor loop完成link到DMA ch0触发下一次ADC通道号的搬运

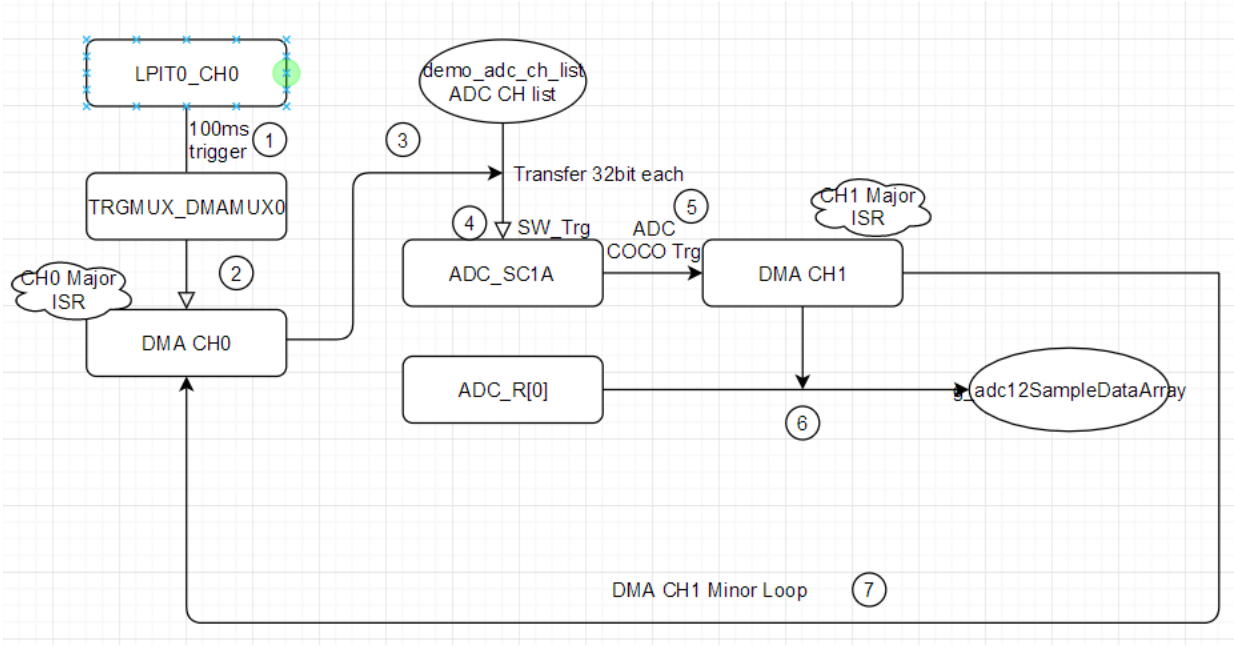
3.这样在下一次LPIT超时到来前就完成所有6个通道的转换。

4.DMA ch0&ch1都设置了Major loop int, 这样一次6个通道转换完成后，两个DMA都会产生中断，表明搬移完成，在主循环中每次完成后打印；

5.这两个示例工程都是采用TCD块，完成DMA设置，并且在转换完成后指定SGA_SLAST还是为这个TCD块，可以不需要在中断中更新DMA设置寄存器；

6.DMA ch0和ch1都没有DREQ，这样每次major loop完成，都无需软件干预都能恢复到可触发状态。所以LPIT可以循环触发。

7.DMA ch0&ch1都设置了Major loop int，这两次Interrupt都不是必要的。ch0的尤其不需要。ch1如果需要在中断中处理数据是需要的。 如果需要定时采集大量的数据，在此之前无需中断，可以将ch1的major int不设置中断，而是去触发另一路DMA ch2, 将此6个数据搬移到另一大buffer。



所以执行序列应该是：1.2.3.4.5.6.7.2.3.4.5.6.7……每个通道重复2,3,4,5,6,7minor loop直至最后一个通道执行Major INT或者Major loop. 本次转换序列结束，等待下一次1的触发。

MULTI_CHANNELS_EDMA_ADC_DMA_SW_TRIG

multi_channels_edma_ADC_DMA_LPIT_HW_TRIG

Overview-multi_channels_edma_ADC_DMA_SW_TRIG 此项目在上面项目的基础上，新增：软件触发功能，

1.通过宏HW_SW_TRIGGER控制是软件触发还是硬件连续触发

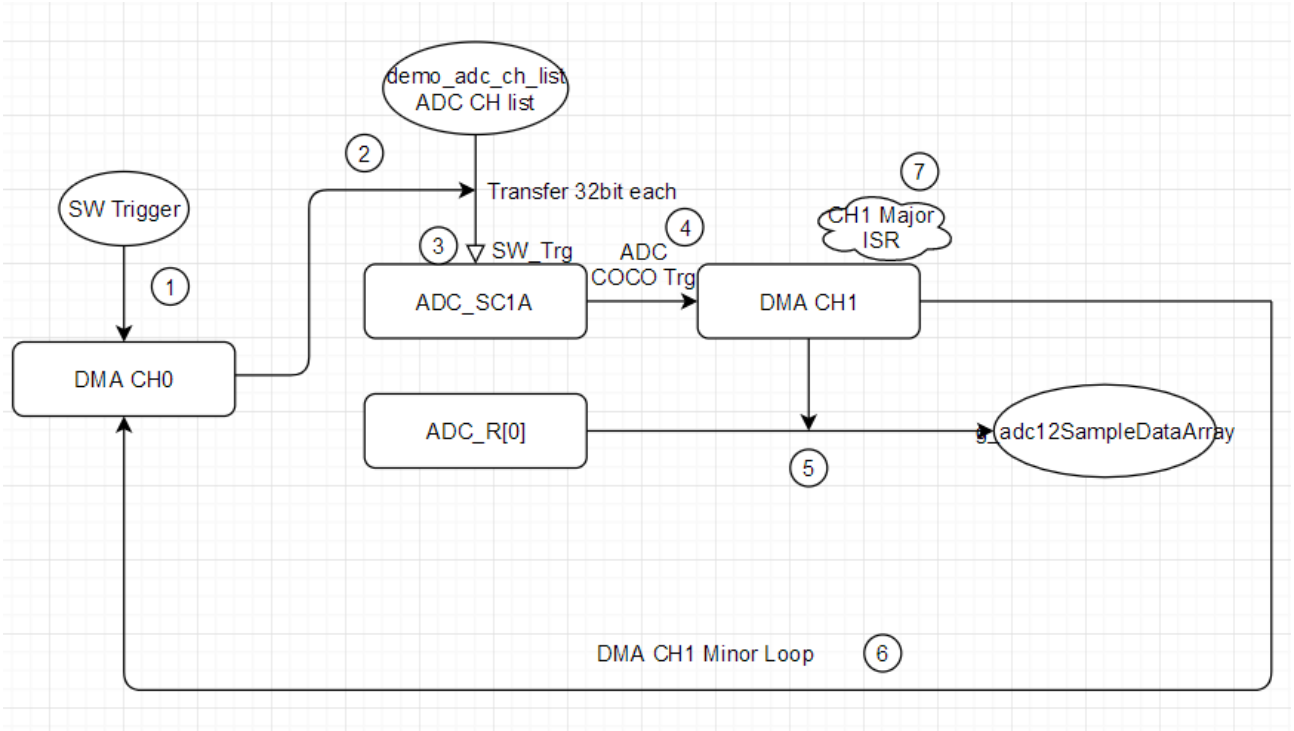
2.软件触发手动触发DMA ch0，以弥补Kinetis ADC缺少连续通道自切换的功能。

3.所以这个功能需要占据：2个DMA通道，至少一个DMA中断

该例程解决的问题主要是弥补Kinetis系列ADC缺少连续通道采样转换，并在结束后产生中断的问题。

执行流程，1,2,3,4,5,6……2,3,4,5,6,7

7为DAM CH1的Major中断，代表所有ADC通道已经转换完成。



END

SECURE CONNECTIONS
FOR A SMARTER WORLD